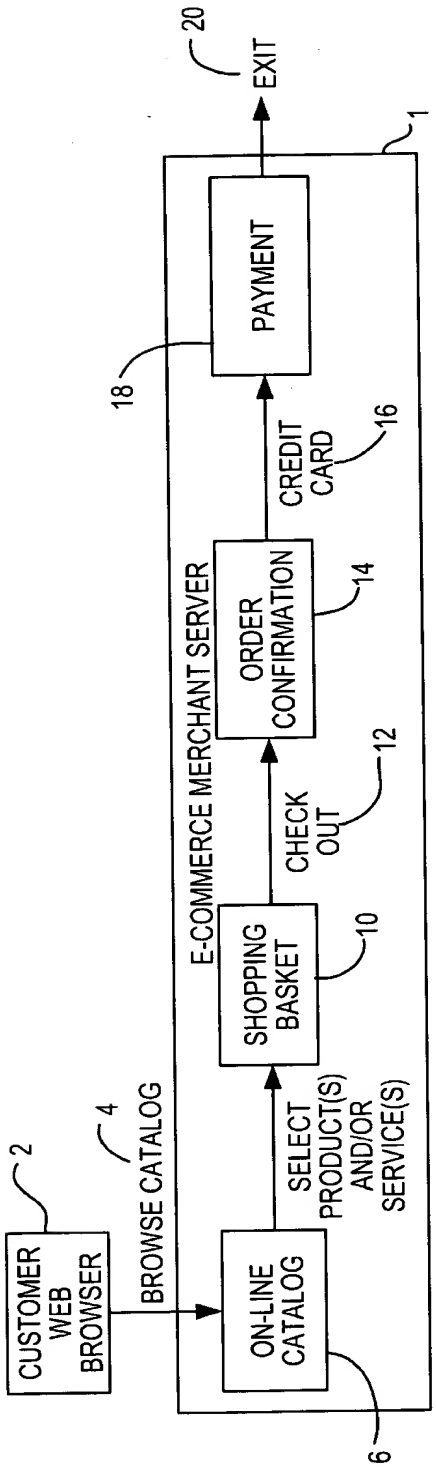

FIG. 1

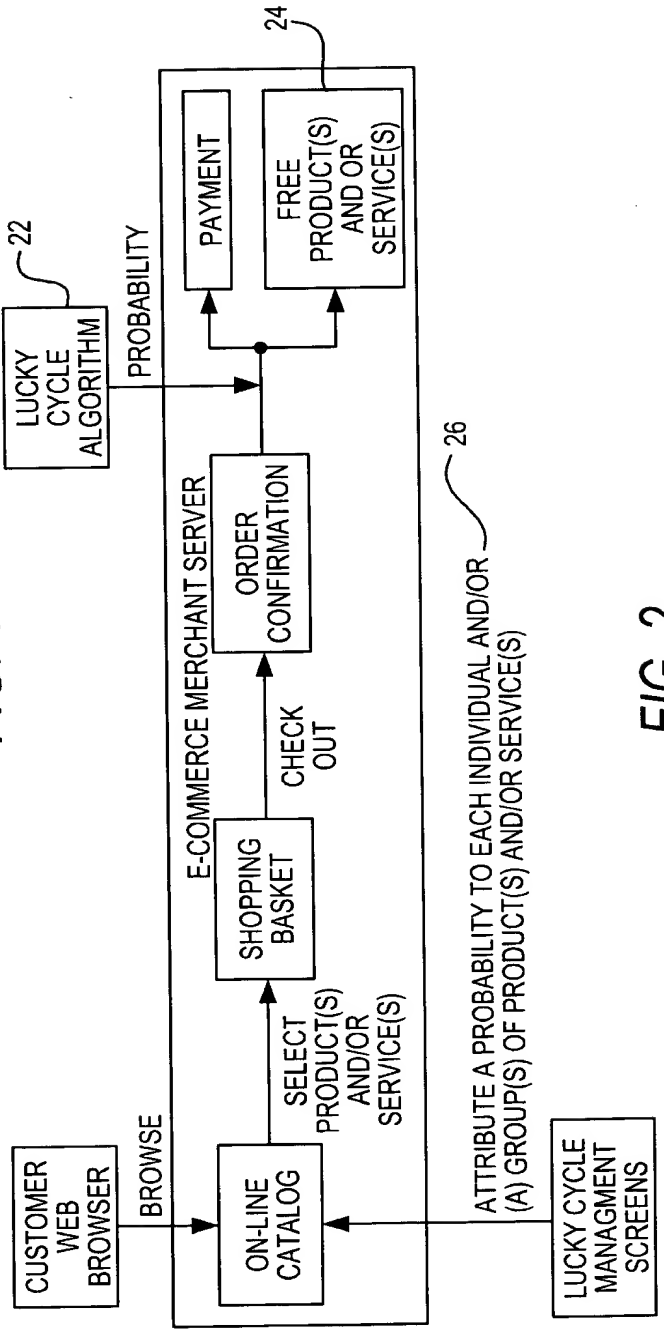

FIG. 2

```
<!--
      Lucky Cycle
      March 2000
      JF MOYERSOEN

      Data Entry Form
-->


<html>
<head>
      <title>Lucky Cycle</title>
</head>

<body>
<center>
<font face=verdana size=3><b>Lucky Cycle</b></font><b>
<form action="result.asp" method=post>
<!-- Display of an Error Message, followed by the initialisation of
this Error Message -->
<br><font face=verdana size=2>
Concept invented and registered by Jean-François Moyersoen
<p> </p>
    <table bgcolor="#E0E0E0" border=1 cellpadding=20 width="461">
      <tr><td>
            <font face=verdana size=2 color=red><i><%=
Session("error_message") %>
            <% Session("error_message") = "" %>
              <table width="419">
            <tr>
                  <td colspan=2><font face=verdana size=2><b>Selected
Algorithm :</b></font></td>
            </tr>
            <tr>
                  <td align=right width="162">
                <input type=radio name="algorithm" value="1" <% If
Session("algorithm") = "1" then response.write(" checked ") %>></td>
                  <td width="245"><font face=verdana size=2>The
regular cycle</font></td>
            </tr>
            <tr>
                  <td align=right width="162">
                <input type=radio name="algorithm" value="2" <% If
Session("algorithm") = "2" then response.write(" checked ") %>></td>
                  <td width="245"><font face=verdana size=2>The
constant probability</font></td>
            </tr>
            <tr>
                  <td align=right width="162">
                <input type=radio name="algorithm" value="3" <% If
Session("algorithm") = "3" then response.write(" checked ") %>></td>
                  <td width="245"><font face=verdana size=2>The
pre-defined list</font></td>
            </tr>
            <tr>
                  <td align=right width="162">
                <input type=radio name="algorithm" value="4" <% If
Session("algorithm") = "4" then response.write(" checked ") %>></td>
                  <td width="245"><font face=verdana size=2>The
dynamic probability</font></td>
```

FIG. 3

```
      </tr>
            <tr>
                  <td colspan=2> </td>
            </tr>
            <tr>
                  <td colspan=2><font face=verdana size=2><b>Parameters
:</b></font></td>
            </tr>
            <tr>
                    <td width="162"><font face=verdana
size=2>Cycle</font></td>
                    <td width="245"><font face=verdana size=2>n=
                  <input type=text name="n" maxlength=3 size=3 value="<%=
Session("n") %>"></font></td>
            </tr>
            <tr>
                    <td width="162"><font face=verdana size=2>Number of
purchases </font></td>
                    <td width="245"><font face=verdana size=2>p=
                  <input maxlength=4 type=text name="pmax" size=3
value="<%= Session("pmax") %>"></font></td>
            </tr>
            <tr>
                  <td colspan=2> </td>
            </tr>
            <tr>
                  <td colspan=2 align=center><input type=submit
value="Simulation"></td>
            </tr>
            </table>
</i></font></td></tr>
</table>
</font></form>
</b></center></body>
</html>
```

FIG. 3A

```
<!-- #include file="algorithm.inc" -->
<%
      '## Input of the form data if the form is not empty
      '## If this page is referred to by a page other than default.asp,
this form does not exist
      '## and the instruction bloc will not be executed
      If Request.form("n") <> "" or Request.Form("pmax") <> "" or
Request.Form("algorithm") <> "" Then
            Session("n") = Trim(Request.Form("n"))
            Session("pmax") = Trim(Request.Form("pmax"))
            Session("algorithm") = Trim(Request.Form("algorithm"))
      End If

      '## Verification of the selected algorithm
      If Session("algorithm") <> "1" and Session("algorithm") <> "2"
and Session("algorithm") <> "3" _
            and Session("algorithm") <> "4" Then Return_Error ("The
algorithm is not correct")

      '## Verification if the value N has been entered
      If Session("n") = "" then Return_Error("N is empty")
      If not Isnumeric(Session("n")) then Return_Error("N is not a
number")
      If Cstr(CLng(Session("n"))) <> Session("n") then Return_Error("N
is not a whole number")
      If CLng(Session("n")) <= 0 Then Return_Error("N must be a
positive number")

      '## Verification of the entered Pmax value
      If Session("pmax") = "" then Return_Error("Pmax is empty")
      If not Isnumeric(Session("pmax")) then Return_Error("Pmax is not
a number")
      If Cstr(CLng(Session("pmax"))) <> Session("pmax") then
Return_Error("Pmax is not a whole number")
      If CLng(Session("pmax")) <= 0 Then Return_Error("Pmax must be a
positive number")

      '## Initialisation of the variables
      nb_articles_won = 0
      Randomize()


      '## Return function to the previous page if an error occurs
      '## the Error Message is stored in the Session("Error_Message")
      Sub Return_Error(p_message)
            Session("Error_Message") = p_message
            response.buffer = true
            response.clear
            response.redirect("default.asp")
            response.end
      End Sub
```

FIG. 4

```
'## Display of the results table
      Sub Table()

            '## Selected algorithm by the Session("algorithm")variable
            Select Case Session("algorithm")

            '## For each algorithm, the index of the ordered article p
varies between 1 and Pmax
            '## For each value p, a function containing the Lucky Cycle
algorithm is called
            '## The parameters to be passed to these different
functions are the cycle n stored in the Session("n") and p
            '## The result is False if the ordered product is not given
for free and True if the product is a free gift

            '## The cell function displays a cell of the table
            '## The parameters to be passed are the index p to be
displayed inside the cell and
            '## the return value of the algorithm that will define the
background color of the cell
            Case "1" :        For p = 1 to Session("pmax")
                                    Cell p,
algorithm_1(Session("n"), p)
                              Next
            Case "2" :        For p = 1 to Session("pmax")
                                    Cell p,
algorithm_2(Session("n"), p)
                              Next
            Case "3" :        For p = 1 to Session("pmax")
                                    Cell p,
algorithm_3(Session("n"), p)
                              Next
            Case "4" :        For p = 1 to Session("pmax")
                                    Cell p,
algorithm_4(Session("n"), p)
                              Next

            End Select

      End Sub

      '## Display of the table cell with a result
      Sub Cell(index_p, reponse_algorithm)

            '## If the cell is the first in a serie of 20, the
following end of line/begin of line tags will be inserted
            if index_p mod 20 = 1 then
                  response.write("</tr><tr>")
            end if

            '## If the index corresponds to a free product, the
background and text color will be defined
            if reponse_algorithm = true then
                  bg_color = "red"
                  text_color = "white"
                  '## The number of articles won is incremented
                  nb_articles_won = nb_articles_won + 1
             else
                 '## If the product is not offered for free, other colors
will be used for the display
```

FIG. 4A

```
                bg_color = "white"
                text_color = "black"
            end if

            '## Display of a cell
            response.write("<td align=center bgcolor='" & bg_color &
"'>" & _
                           "<font color='" & text_color & "' face=verdana
size=2>" & index_p & "</td>")

      End Sub

%>
<html>
<head>
      <title>Lucky Cycle</title>
</head>

<body>
<table cellspacing=1 cellpadding=3>
<tr>
      <td colspan=20><font face=verdana size=2><b>Result
Table</b></font></td>
<% Call Table %>
</tr>
<tr>
      <td colspan=20><br><font face=verdana size=2><b><%=
nb_articles_won %> articles on <%= Session("pmax") %> have been won

      <%    '## If the number of articles is different from zero
            If nb_articles_won <> 0 Then %>
            (1 on <%= FormatNumber(Session("pmax")/nb_articles_won,3)
%>)
      <%    End If %>
      <br></b> Theoretical Cycle = <%= Session("n") %>
      <br><br>
      <form action=result.asp method=post>
      <input type=button value="Back"
onclick="document.location.href='default.asp'">
      <input type=submit value="New Simulation">
      </form>
      </font></td>
</tr>
</table>

</body>
</html>
```

FIG. 4B

```
<%

Dim p_won    '## Variable storing the index of the next article that
will be offered free
                          '## (or that will be used as a reference for
the dynamic probability algorithm)

'## All the procedures use the parameters cycle n (cycle_n) and the
index p (index_p)
'## The result of each procedure is a boolean (True if the article is
given free or False in the other situation)

'## The regular cycle
'## is based on a fixed cycle : after (n-1) articles have been sold,
the nth article is offered free
'## Mathematically, it could be stated that the article is offered free
when
'## index_p Mod cycle_n = constant number between 0 and (n-1)
'## For example : if index_p Mod cycle_n = 0

Function Algorithm_1(cycle_n, index_p)
      If index_p Mod cycle_n = 0 Then
            Algorithm_1 = True
      Else
            Algorithm_1 = False
      End If
End Function

'## The constant probability
'## The cycle is based on a constant probability of 1/n
'## Mathematically, this cycle is characterized by the generation of a
random number between 0 and (n-1)
'## If this number equals any constant between 0 and (n-1), then the
article is offered free
'## For example, if the number is equal to 0

Function Algorithm_2(cycle_n, index_p)
      nb_random = Int(cycle_n * Rnd)
      If nb_random = 0 Then
            Algorithm_2 = True
      Else
            Algorithm_2 = False
      End If
End Function
```

FIG. 5

'## The pre-defined list

```
'## This cycle is characterized by the creation of a predefined list
with all the indexes p that will be future winners
'## This list will be created on regular intervals, depending on the
number of elements defined in the list
'## This list must itself respect the cycle n and as a result the
probability 1/n.
'## The algorithm underneath represents a special case in which the
list contains only one element
'## and is thus rebuild every time n articles have been ordered
'## In this situation, this list is created by randomly assigning a
number between index_p and index_p + cycle_n

Function Algorithm_3(cycle_n, index_p)
      '## Creation of the list if the article of the index p begins
with a serie of n orders
      '## this means if the index_p mod cycle_n = 1
      '## Special case : if the cycle_n = 1 then no matter what the
value is of p,
      '## a list will be recreated (the article is the first of a serie
of 1 order), when p mod 1 <> 1
      If index_p mod cycle_n = 1 or cycle_n=1 Then
            p_won = index_p + Int(cycle_n * Rnd)
      End If
      '## If the index p is found in the list p_won containing a single
element, it will be offered free
      If index_p = p_won Then
            Algorithm_3 = True
      Else
            Algorithm_3 = False
      End If
End Function
```

'## The dynamic probability

```
'## This cycle calculates the probability of an order with index p in
function of a winning reference order,
'## that in this case would correspond to a regular cycle (see the
first algorithm)
'## The probability is calculated in function of the index_p and the
winning reference order
'## In the function underneath, we take as a reference list (n, 2*n,
3*n, 4*n, ...)
'## This list can contain any value as long as it respects itself the
cycle n and the probability 1/n

Function Algorithm_4(cycle_n, index_p)
      '## Initialisation during the first passage of p_won = cycle_n
      If index_p = 1 Then
            p_won = cycle_n
      End If
      '## Calculation of the inverse of the probability
      '## In this case, we take (p_won - index_p + 1)
      Inv_probability = (p_won - index_p + 1)
      '## Generation of a random number between 0 and
(inv_probability -1)
     nb_random = Int(Inv_probability * Rnd)
     '## If the number is equal to 0, the product is offered free
     If nb_random = 0 Then
           Algorithm_4 = True
```

FIG. 5A

```
            '## The winning reference order takes the following index
of the reference list
            p_won = CInt(p_won) + CInt(cycle_n)
      Else
            Algorithm_4 = False
      End If
End Function
%>
```

FIG. 5B

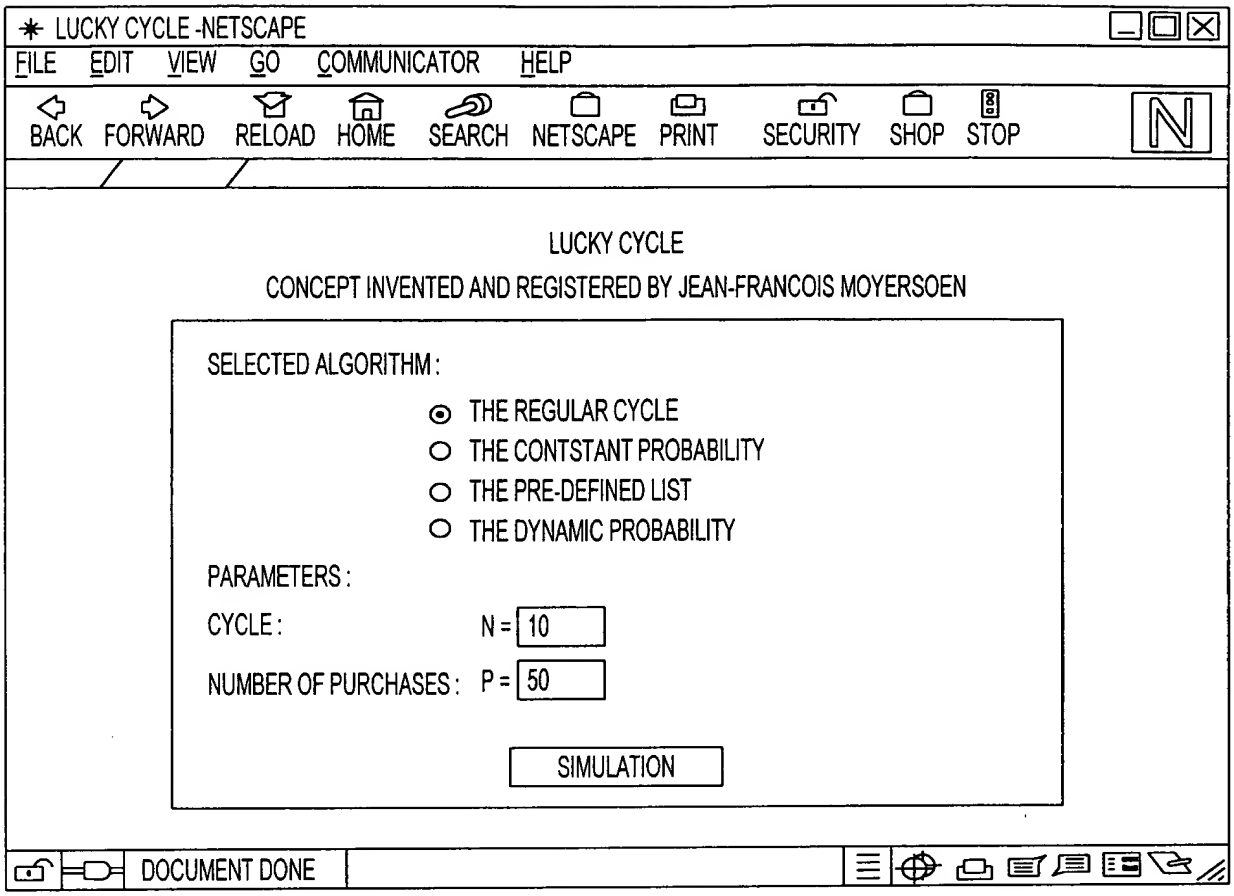
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
LUCKY CYCLE
CONCEPT INVENTED AND REGISTERED BY JEAN-FRANCOIS MOYERSOEN
SELECTED ALGORITHM:
THE REGULAR CYCLE
THE CONTSTANT PROBABILITY
THE PRE-DEFINED LIST
THE DYNAMIC PROBABILITY
PARAMETERS:
CYCLE: N = 10
NUMBER OF PURCHASES: P = 50
SIMULATION
DOCUMENT DONE

FIG. 6

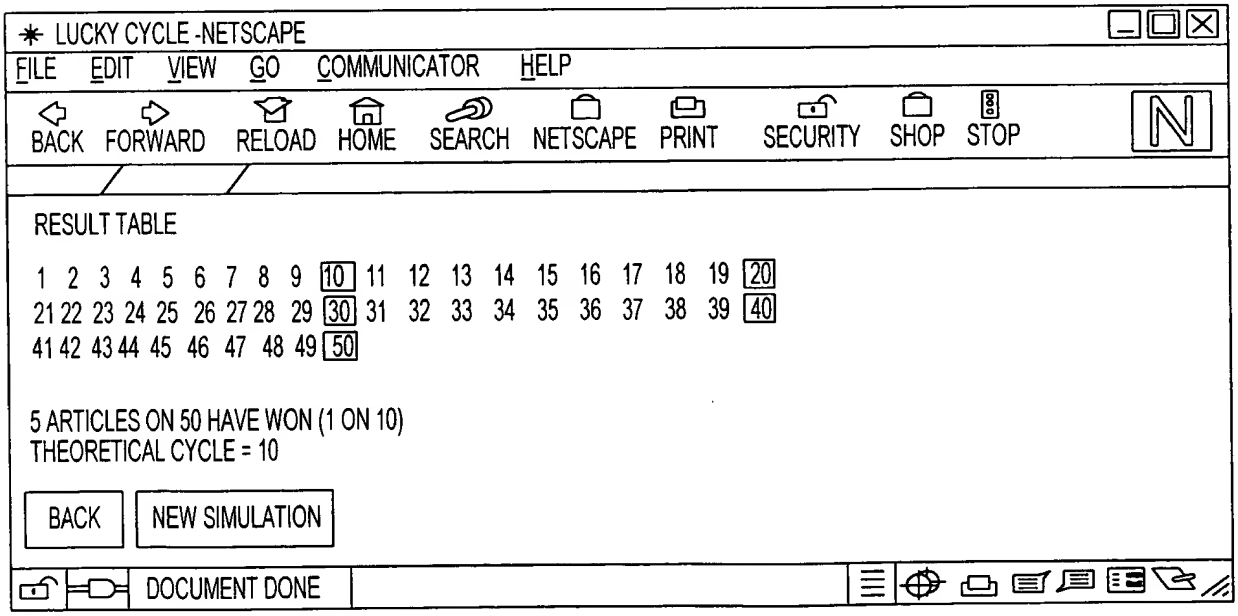
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
RESULT TABLE
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
5 ARTICLES ON 50 HAVE WON (1 ON 10)
THEORETICAL CYCLE = 10
BACK
NEW SIMULATION
DOCUMENT DONE

FIG. 7

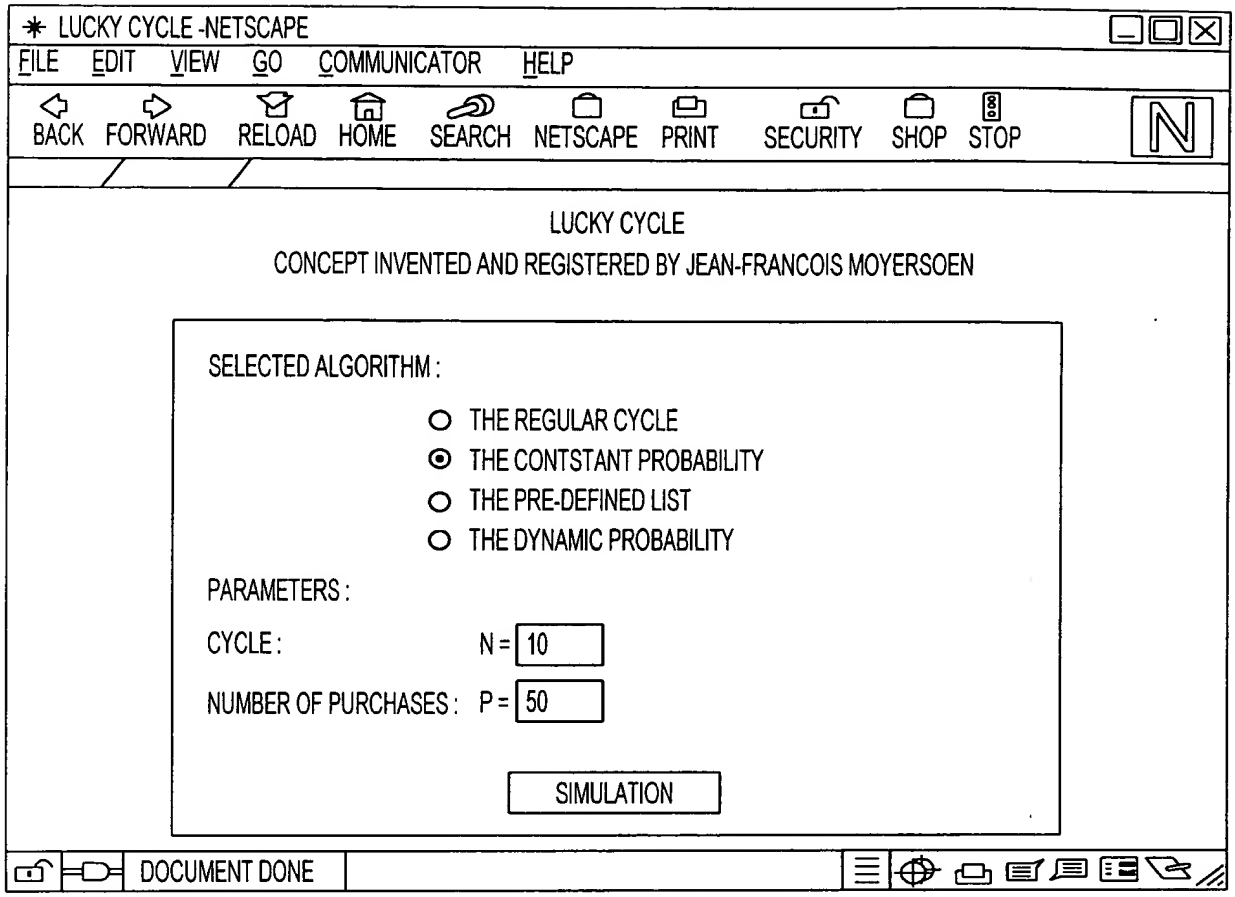
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
LUCKY CYCLE
CONCEPT INVENTED AND REGISTERED BY JEAN-FRANCOIS MOYERSOEN
SELECTED ALGORITHM :
THE REGULAR CYCLE
THE CONTSTANT PROBABILITY
THE PRE-DEFINED LIST
THE DYNAMIC PROBABILITY
PARAMETERS :
CYCLE : N = 10
NUMBER OF PURCHASES : P = 50
SIMULATION
DOCUMENT DONE

FIG. 8

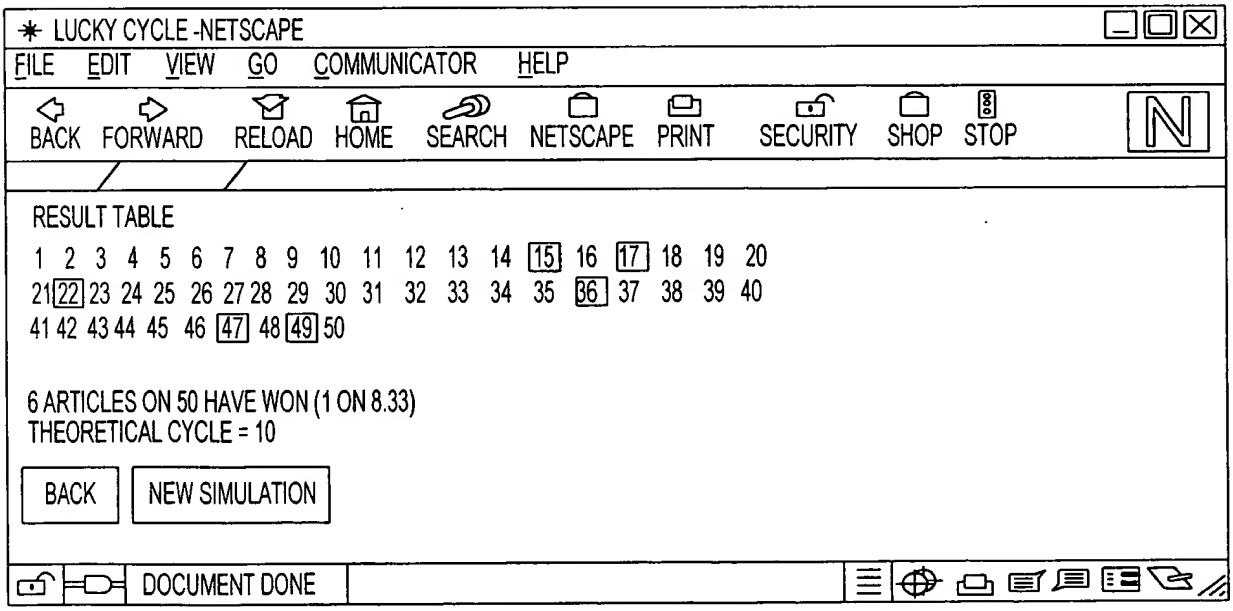
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
RESULT TABLE
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
6 ARTICLES ON 50 HAVE WON (1 ON 8.33)
THEORETICAL CYCLE = 10
BACK
NEW SIMULATION
DOCUMENT DONE

FIG. 9

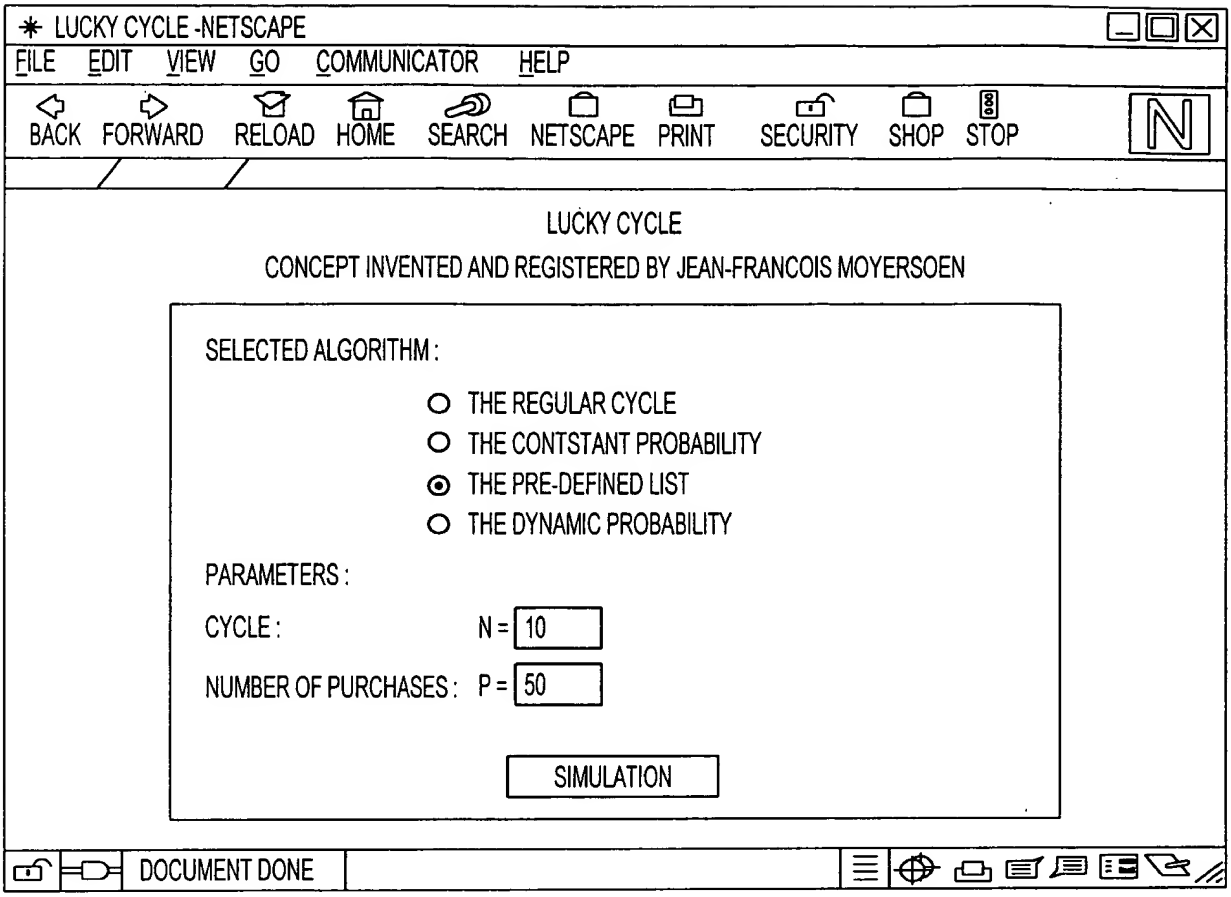
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
LUCKY CYCLE
CONCEPT INVENTED AND REGISTERED BY JEAN-FRANCOIS MOYERSOEN
SELECTED ALGORITHM :
THE REGULAR CYCLE
THE CONTSTANT PROBABILITY
THE PRE-DEFINED LIST
THE DYNAMIC PROBABILITY
PARAMETERS :
CYCLE : N = 10
NUMBER OF PURCHASES : P = 50
SIMULATION
DOCUMENT DONE

FIG. 10

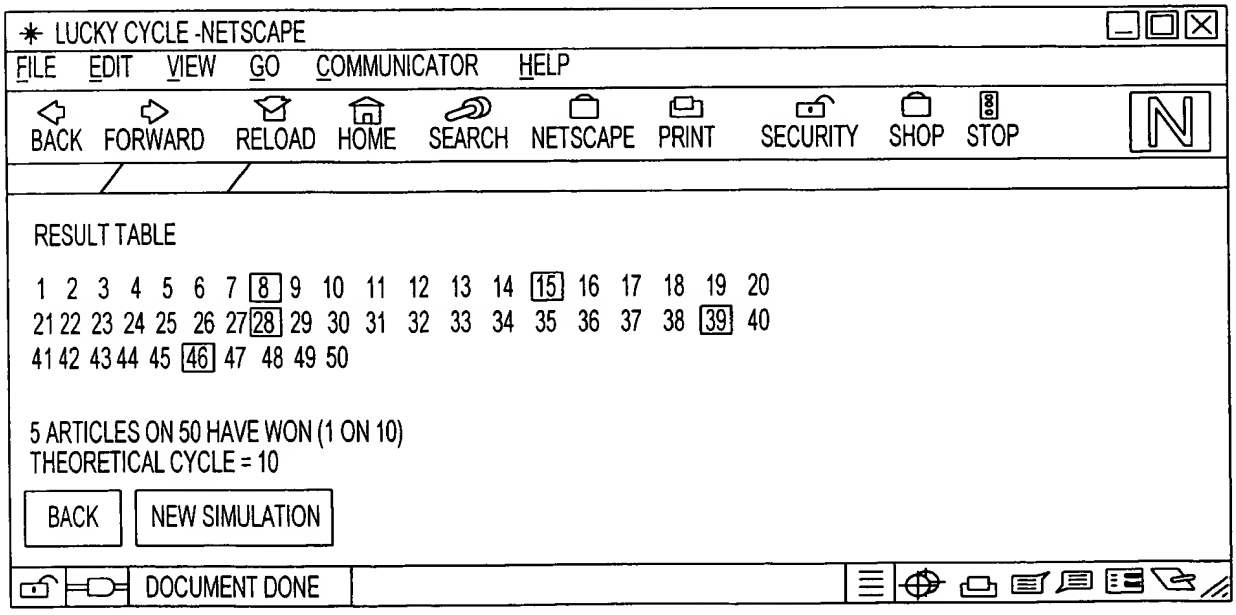
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
RESULT TABLE
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
5 ARTICLES ON 50 HAVE WON (1 ON 10)
THEORETICAL CYCLE = 10
BACK
NEW SIMULATION
DOCUMENT DONE

FIG. 11

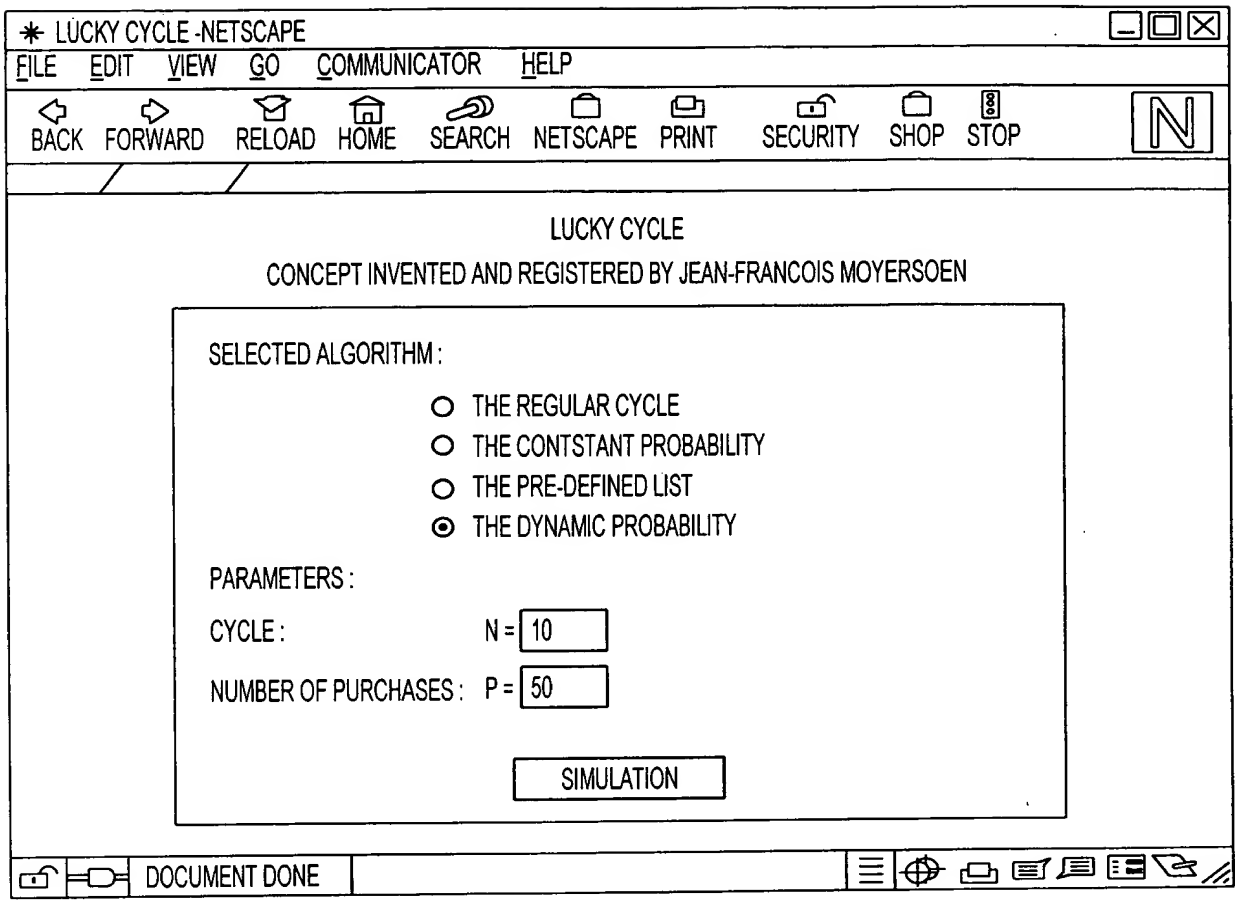
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
LUCKY CYCLE
CONCEPT INVENTED AND REGISTERED BY JEAN-FRANCOIS MOYERSOEN
SELECTED ALGORITHM :
THE REGULAR CYCLE
THE CONTSTANT PROBABILITY
THE PRE-DEFINED LIST
THE DYNAMIC PROBABILITY
PARAMETERS :
CYCLE : N = 10
NUMBER OF PURCHASES : P = 50
SIMULATION
DOCUMENT DONE

FIG. 12

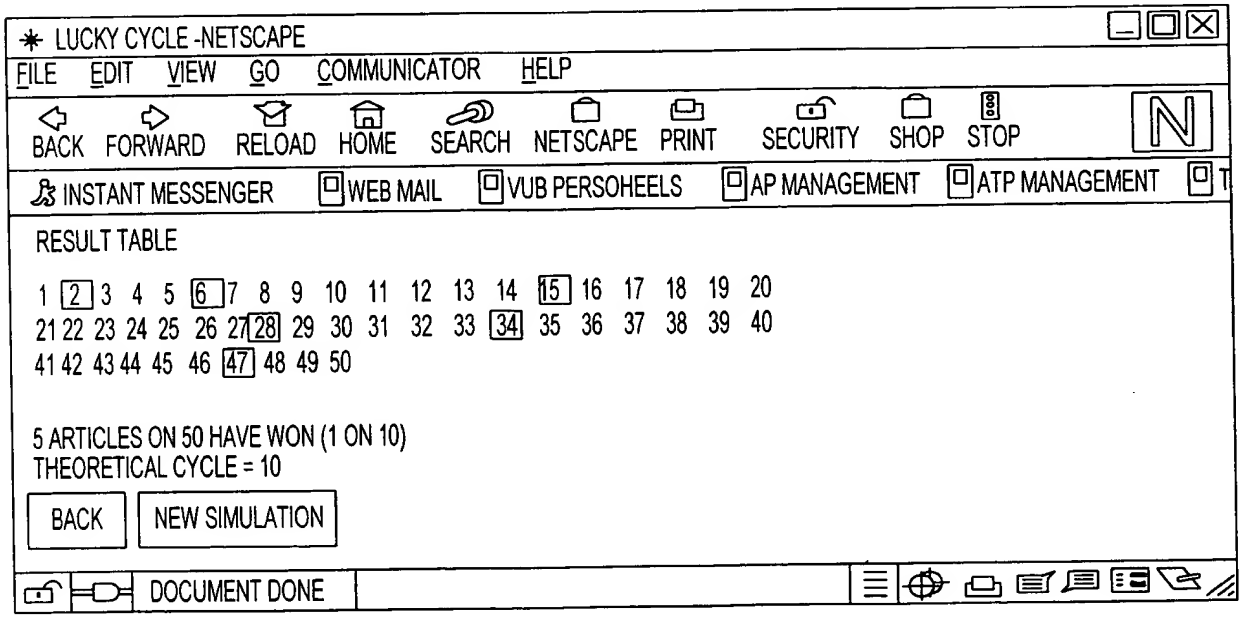
LUCKY CYCLE -NETSCAPE
FILE EDIT VIEW GO COMMUNICATOR HELP
BACK FORWARD RELOAD HOME SEARCH NETSCAPE PRINT SECURITY SHOP STOP
INSTANT MESSENGER WEB MAIL VUB PERSOHEELS AP MANAGEMENT ATP MANAGEMENT
RESULT TABLE
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
5 ARTICLES ON 50 HAVE WON (1 ON 10)
THEORETICAL CYCLE = 10
BACK
NEW SIMULATION
DOCUMENT DONE

FIG. 13